Steps To Use Puppet Code

Objective: Steps required to apply puppet code to

- Automate Jenkins installation with Puppet
- Ensure required dependencies (Java 21, fontconfig) are installed
- Configure Jenkins to run on custom port (8000)
- Enable Jenkins service
- Ensure idempotency so repeated runs do not cause errors or duplicate configurations

Steps To Use Puppet Code

1. **Create Puppet Directory Structure for jenkins on puppet server**

create

- jenkins/manifests directory inside **/etc/puppetlabs/code/environments/production/modules/** directory
- **init.pp** file inside manifests directory

```
[root@master modules]# pwd
/etc/puppetlabs/code/environments/production/modules
[root@master modules]# ls -ltr
total 0
[root@master modules]# mkdir jenkins
[root@master modules]# ls -ltr
total 0
drwxr-xr-x 2 root root 6 Jan  3 16:23 jenkins
[root@master modules]# cd jenkins
[root@master jenkins]# mkdir manifests
[root@master jenkins]# cd manifests/
[root@master manifests]# ls
[root@master manifests]# touch init.pp
[root@master manifests]# ls -ltr
total 0
-rw-r--r-- 1 root root 0 Jan  3 16:24 init.pp
[root@master manifests]# pwd
/etc/puppetlabs/code/environments/production/modules/jenkins/manifests
[root@master manifests]#
[root@master manifests]# tree /etc/puppetlabs/code/environments/production/modules/jenkins
/etc/puppetlabs/code/environments/production/modules/jenkins
└── manifests
    └── init.pp

1 directory, 1 file
[root@master manifests]#
```

Steps To Use Puppet Code

2. Put the PuppetCode inside the init.pp file on puppet server

- Copy the code from **“PuppetCode”** file and past it inside the **init.pp** file.
- PuppetCode file is shared as an attachment

```
[root@master manifests]# pwd
/etc/puppetlabs/code/environments/production/modules/jenkins/manifests
[root@master manifests]# ls
init.pp
[root@master manifests]# cat init.pp
[root@master manifests]# vi init.pp
[root@master manifests]#
[root@master manifests]# cat init.pp
#Puppet code to install jenkins on CentOS 9
class jenkins {
# ----------------------------
# Jenkins repository
# ----------------------------
yumrepo { 'jenkins':
baseurl => 'https://pkg.jenkins.io/redhat-stable',
descr => 'Jenkins Stable Repo',
enabled => 1,
gpgcheck => 1,
gpgkey => 'https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key',
}
# ----------------------------
# Adoptium repository
# ----------------------------
yumrepo { 'adoptium':
baseurl => 'https://packages.adoptium.net/artifactory/rpm/rhel/$releasever/$basearch',
descr => 'Adoptium',
enabled => 1,
gpgcheck => 1,
gpgkey => 'https://packages.adoptium.net/artifactory/api/gpg/key/public',
}
# ----------------------------
# Required packages
```

Steps To Use Puppet Code

3. Node Declaration in site.pp on puppet server

- Apply the jenkins class on node agent01.puppet.io inside site.pp file
- Node name is **the hostname of the Puppet agent**.
- **Replace `agent01.puppet.io` with the actual agent hostname, in your environment.**

```
[root@master manifests]# pwd
/etc/puppetlabs/code/environments/production/manifests
[root@master manifests]# ll
total 4
-rw-r--r-- 1 root root 48 Jan  3 13:48 site.pp
[root@master manifests]# cat site.pp
node 'agent01.puppet.io' {
  include jenkins
}

[root@master manifests]#
```

Your agent hostname

Steps To Use Puppet Code

4. Puppet Agent Configuration

- Puppet agent must be configured to communicate with the Puppet server
- This is done in the agent's puppet.conf file
- Server -> Pupper master hostname
- Environment -> Puppet environment (e.g. production)
- Certname -> Puppet agent hostname

```
[root@agent01 ~]# cat /etc/puppetlabs/puppet/puppet.conf | grep -v "#"
[main]
certname = agent01.puppet.io
server = master.puppet.io
environment = production
runinterval = 1h
[root@agent01 ~]#
[root@agent01 ~]#
```

Note : Replace above parameter values(server,environment,certname as per your environment

Steps To Use Puppet Code

5. Apply the Puppet Code on Puppet Agent

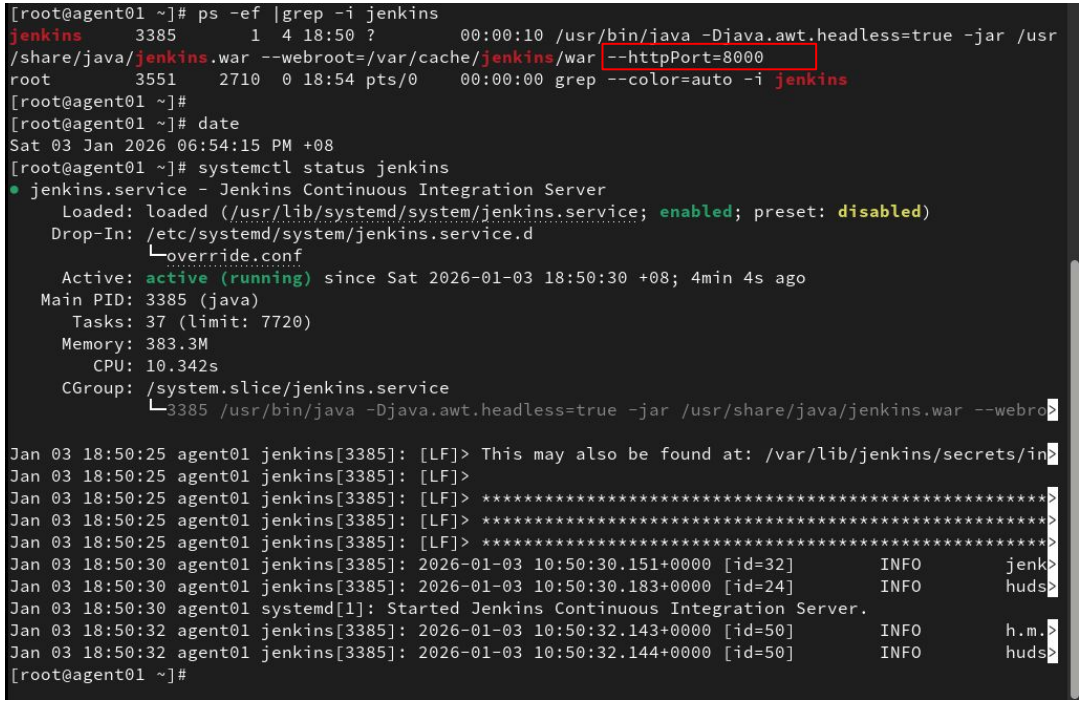
- Run puppet agen -t to apply Puppet Code on Agent

```
[root@agent01 ~]# puppet agent -t
Info: Using environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Caching catalog for agent01.puppet.io
Info: Applying configuration version '1767437414'
Notice: /Stage[main]/Jenkins/Yumrepo[jenkins]/ensure: created (corrective)
Info: Yumrepo[jenkins](provider=inifile): changing mode of /etc/yum.repos.d/jenkins.repo from 600 to
 644
Notice: /Stage[main]/Jenkins/Yumrepo[adoptium]/ensure: created (corrective)
Info: Yumrepo[adoptium](provider=inifile): changing mode of /etc/yum.repos.d/adoptium.repo from 600
to 644
Notice: /Stage[main]/Jenkins/Package[jenkins]/ensure: created
Notice: /Stage[main]/Jenkins/Service[jenkins]/ensure: ensure changed 'stopped' to 'running'
Info: /Stage[main]/Jenkins/Service[jenkins]: Unscheduling refresh on Service[jenkins]
Notice: Applied catalog in 25.96 seconds
[root@agent01 ~]#
```

Steps To Use Puppet Code

6. Outcome

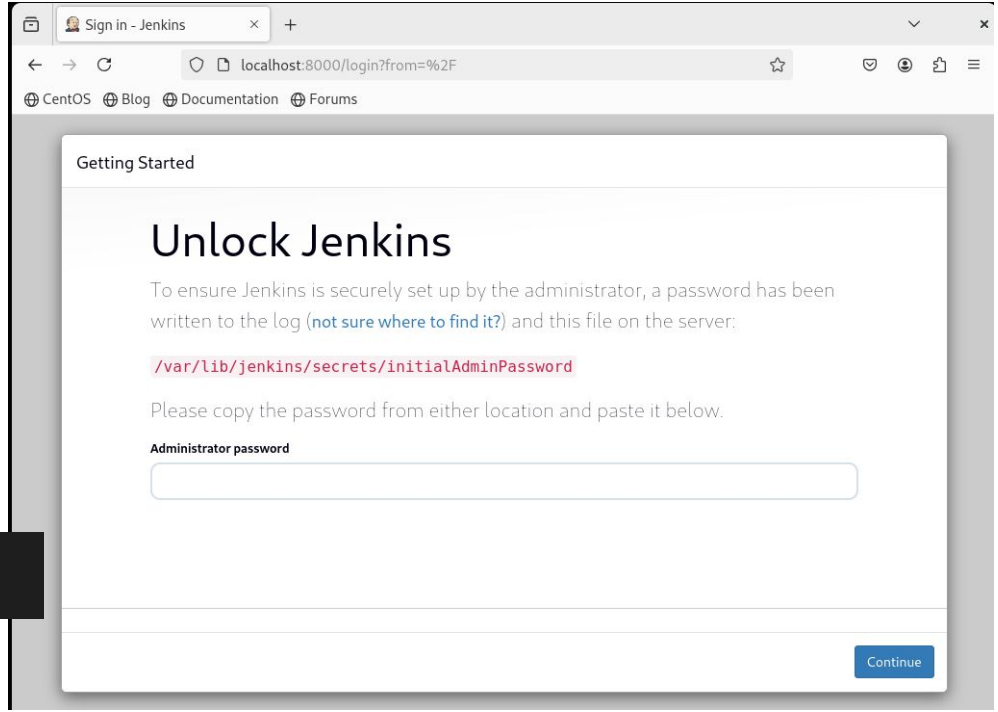
- Jenkins installed and configured
- Firewall port opened
- Jenkins service running on port 8000



Steps To Use Puppet Code

7. Verification

- Open browser and access jenkins UI
- localhost:8000
- rpm -qa command to verify jenkins packages installed successfully




```
[root@agent01 ~]# rpm -qa --last|grep -i jenkins
jenkins-2.528.3-1.1.noarch                    Sat 03 Jan 2026 06:50:20 PM
[root@agent01 ~]#
```

Steps To Use Puppet Code

8. Check Idempotency

- Run the puppet agent -t to ensure idempotency so repeated runs do not cause errors or duplicate configurations